

CREATE

(`0` :Persona {nombre:'Danny',apellido:'Díaz'}) ,

(`3` :Persona {nombre:'William',apellido:'Astudillo'}) ,

(`4` :Persona {nombre:'Néstor',apellido:'Alarcón'}) ,

(`6` :Persona {nombre:'Esteban',apellido:'Guamán'}) ,

(`7` :Universidad {siglas:'EPN'}) ,

(`8` :Universidad {siglas:'ESPOCH'}) ,

(`9` :Universidad {siglas:'ESPE'}) ,

(`10` :Empresa {nombre:'Grafinsa'}) ,

(`11` :Hobbie {tema:'Crear videojuegos'}) ,

(`12` :Ciudad {nombre:'Quito'}) ,

(`13` :`Categoría` {nombre:'A'}) ,

(`14` :`Categoría` {nombre:'B'}) ,

(`0`)-[:`es\_amigo\_de` ]->(`3`),

(`0`)-[:`es\_amigo\_de` ]->(`4`),

(`0`)-[:`es\_amigo\_de` ]->(`6`),

(`3`)-[:`es\_amigo\_de` ]->(`0`),

(`4`)-[:`es\_amigo\_de` ]->(`0`),

(`6`)-[:`es\_amigo\_de` ]->(`0`),

(`0`)-[:`RELATED\_TO` ]->(`0`),

(`0`)-[:`estudia\_en` ]->(`7`),

(`3`)-[:`RELATED\_TO` ]->(`4`),

(`3`)-[:`RELATED\_TO` ]->(`4`),

(`3`)-[:`estudia\_en` ]->(`8`),

(`4`)-[:`estudia\_en` ]->(`9`),

(`6`)-[:`trabaja\_en` ]->(`10`),

(`0`)-[:`tiene\_el\_hobbie` ]->(`11`),

(`6`)-[:`tiene\_el\_hobbie` ]->(`11`),

(`0`)-[:`vive\_en` ]->(`12`),

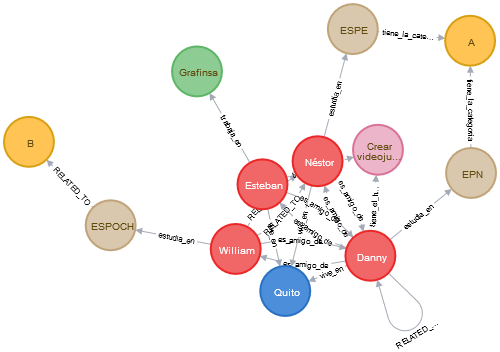
(`4`)-[:`vive\_en` ]->(`12`),

(`6`)-[:`vive\_en` ]->(`12`),

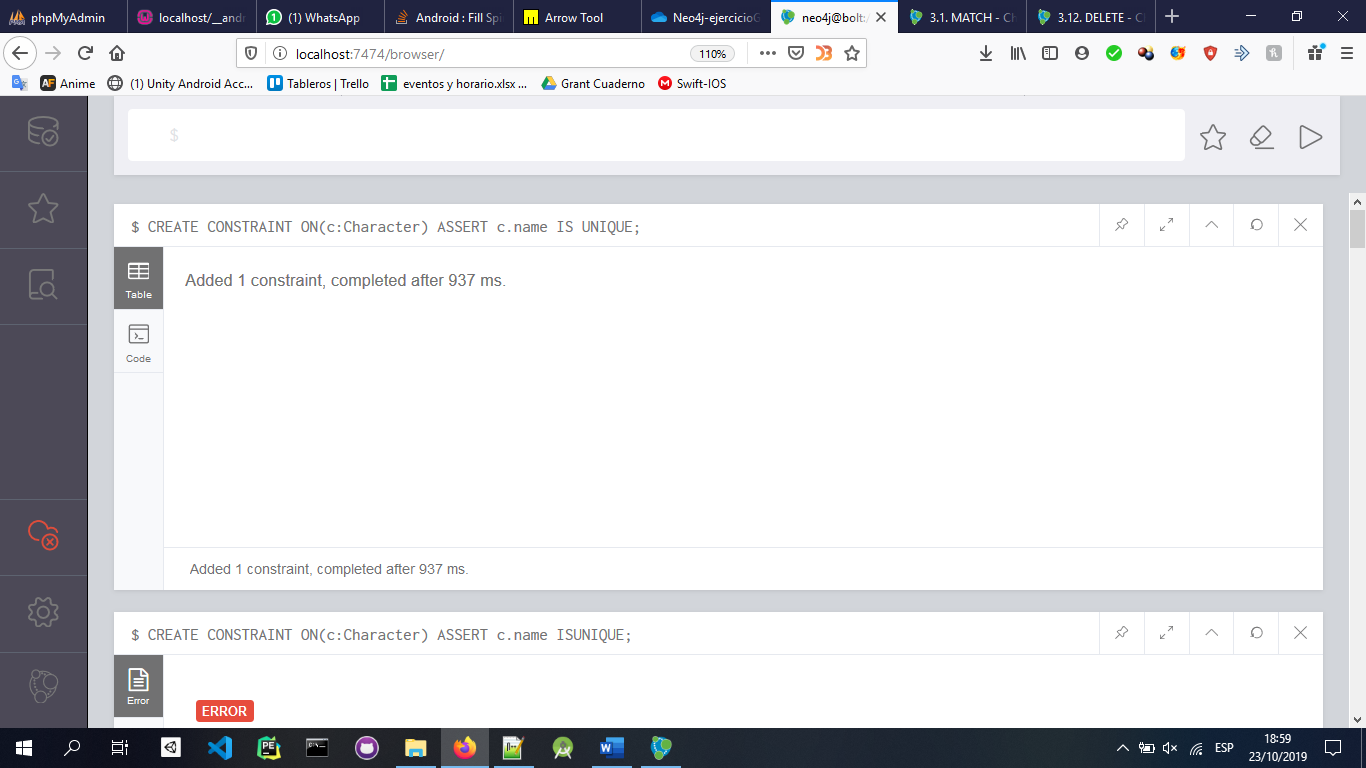
(`7`)-[:`tiene\_la\_categoria` ]->(`13`),

(`9`)-[:`tiene\_la\_categoria` ]->(`13`),

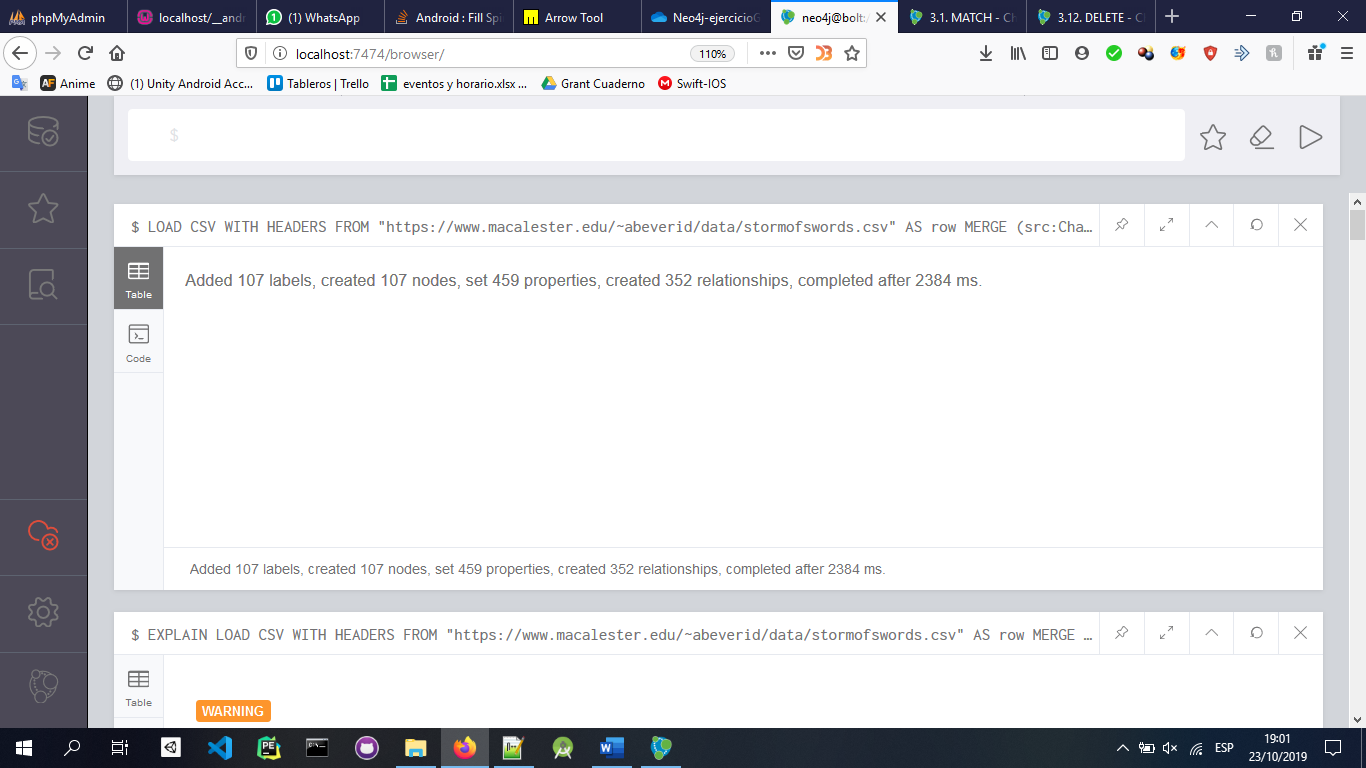
(`8`)-[:`RELATED\_TO` ]->(`14`)



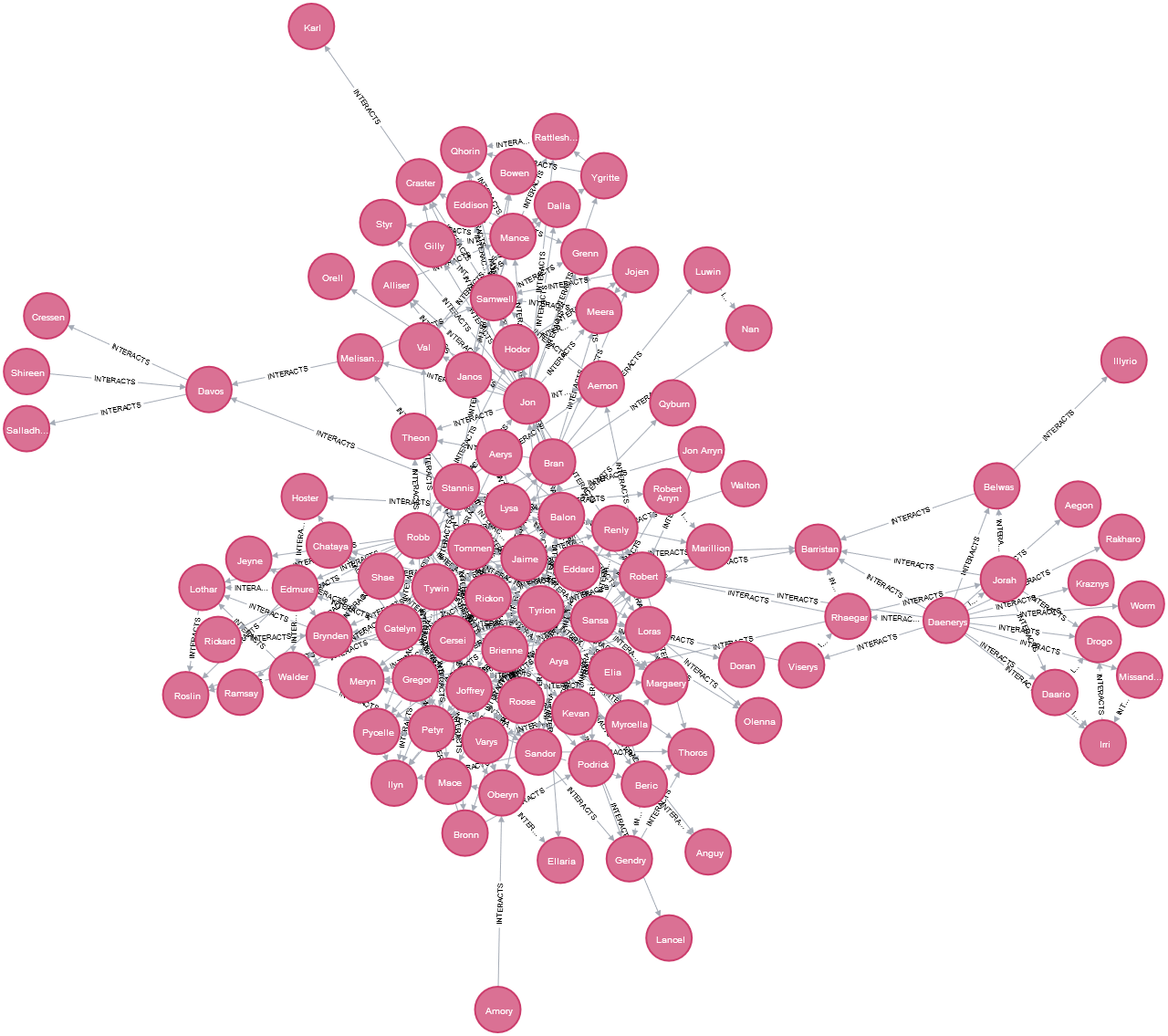
CREATE CONSTRAINT ON(c:Character) ASSERT c.name IS UNIQUE;



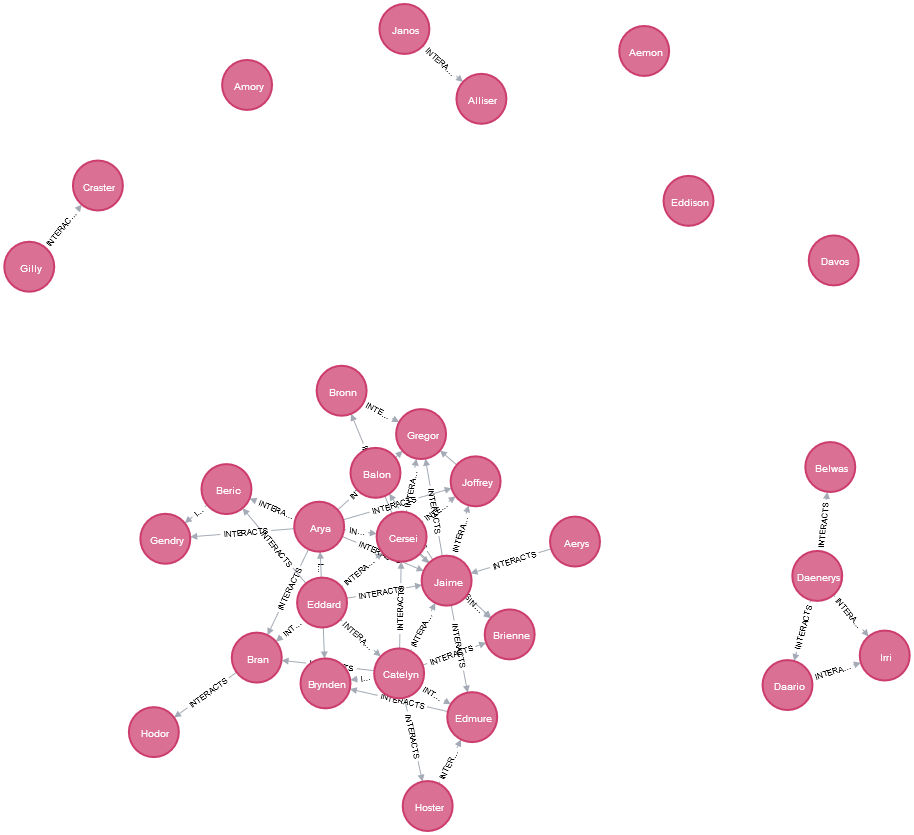
LOAD CSV WITH HEADERS FROM "https://www.macalester.edu/~abeverid/data/stormofswords.csv" AS row MERGE (src:Character{name: row.Source}) MERGE(tgt:Character{name: row.Target}) MERGE(src)-[r:INTERACTS]->(tgt) ON CREATE SET r.weight=toInteger(row.Weight)

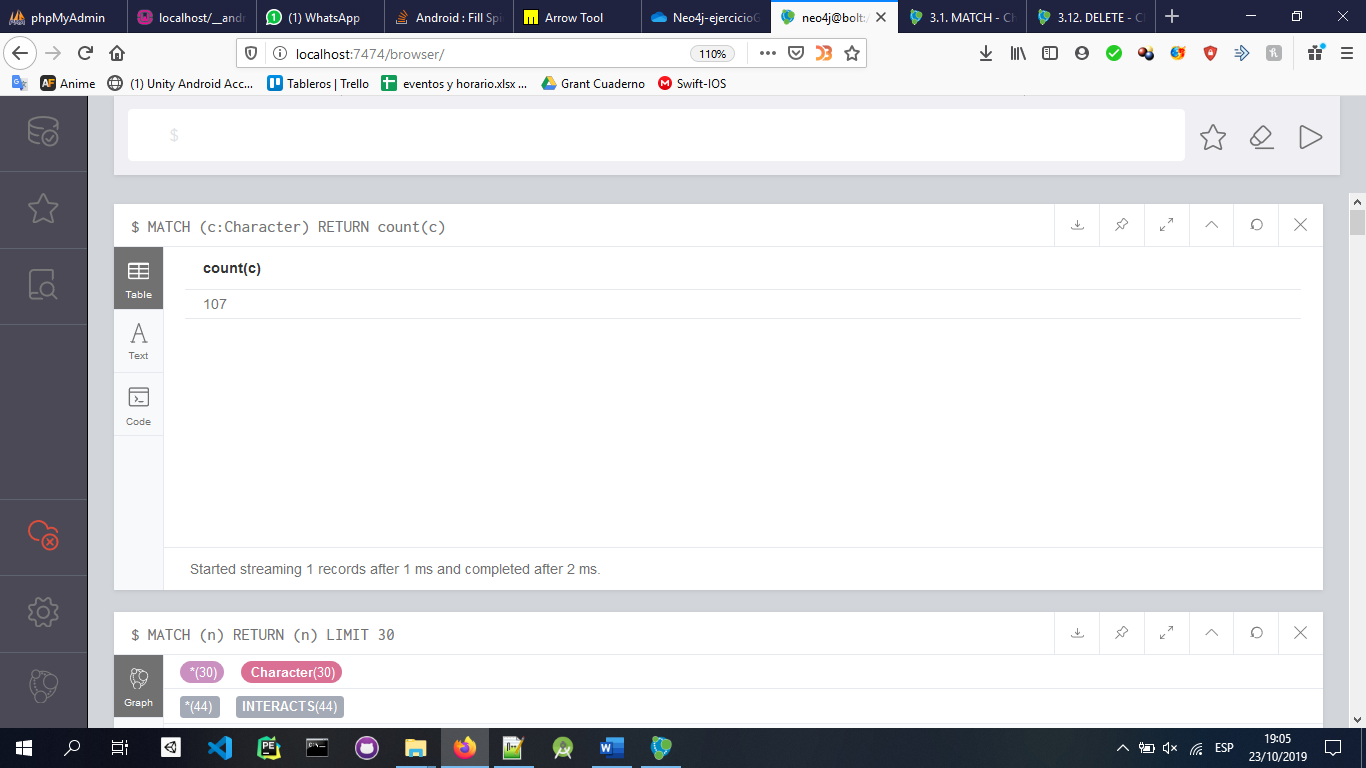


MATCH p=(:Character)-[:INTERACTS]-(:Character) RETURN p

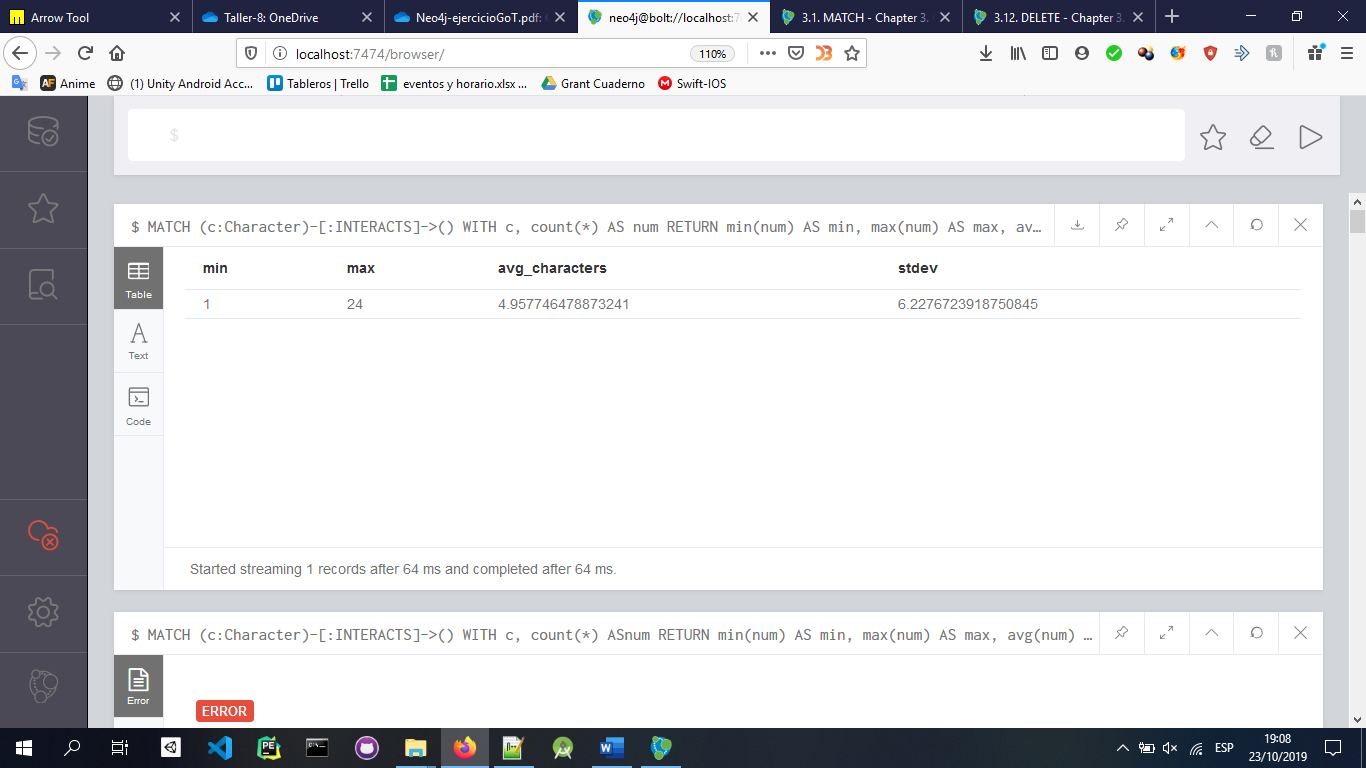


MATCH (n) RETURN (n) LIMIT 30

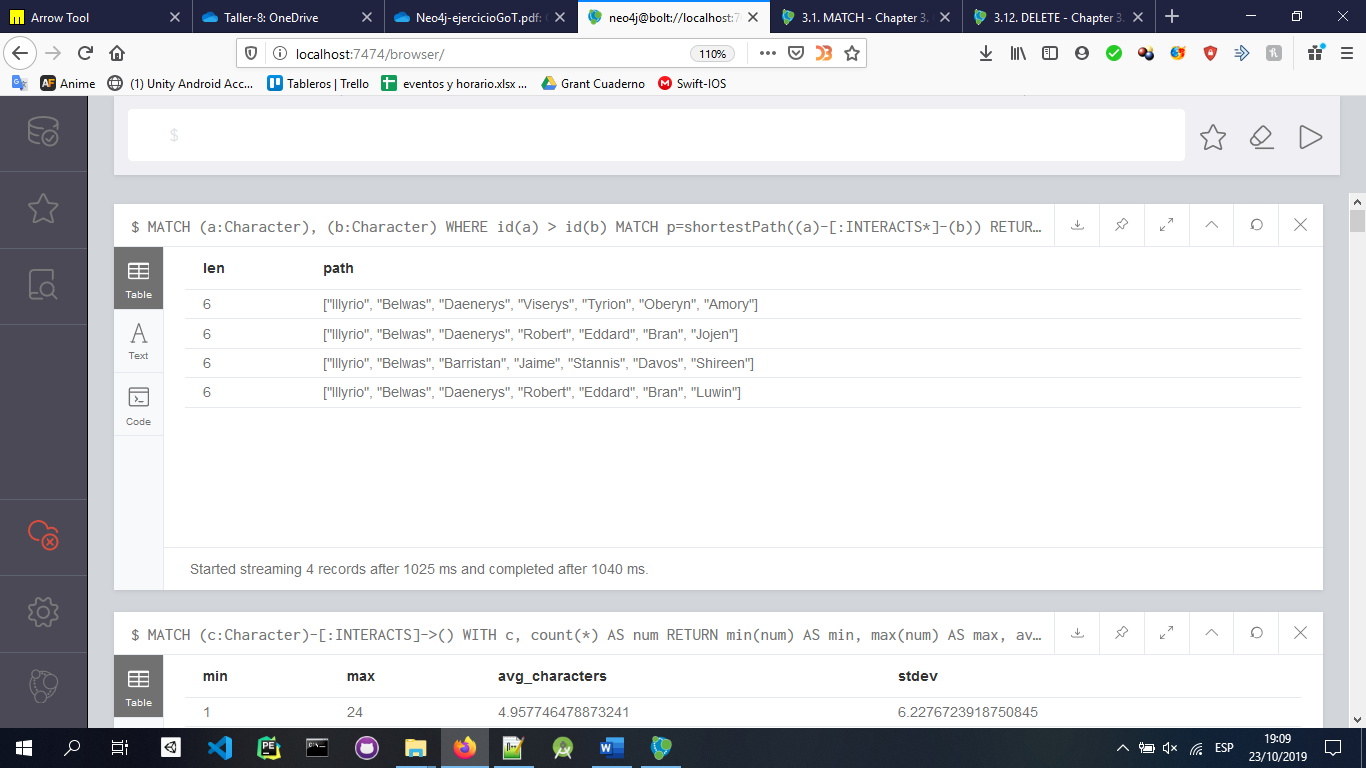
MATCH (c:Character) RETURN count(c)

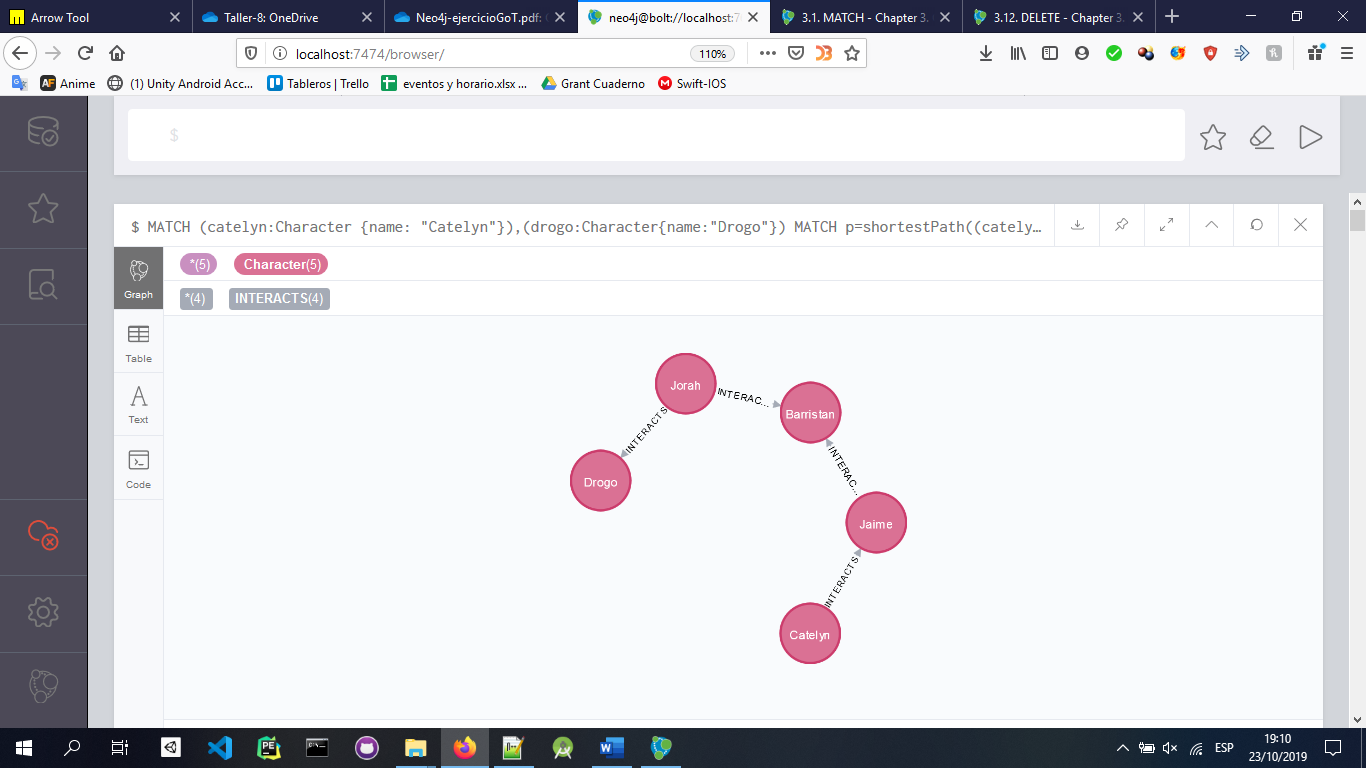


MATCH (c:Character)-[:INTERACTS]->() WITH c, count(\*) AS num RETURN min(num) AS min, max(num) AS max, avg(num) AS avg\_characters, stdev(num) AS stdev

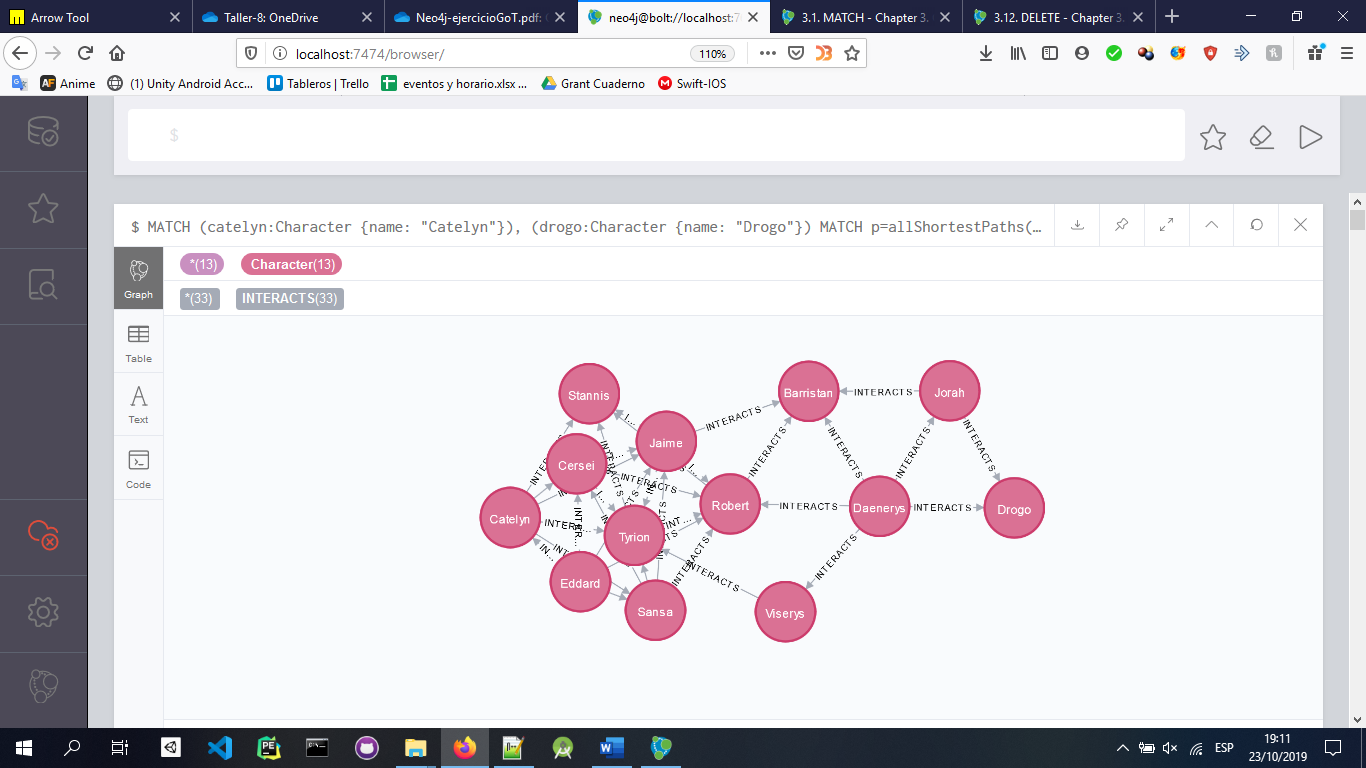


MATCH (a:Character), (b:Character) WHERE id(a) > id(b) MATCH p=shortestPath((a)-[:INTERACTS\*]-(b)) RETURN length(p) AS len, extract(x IN nodes(p) | x.name) AS path ORDER BY len DESC LIMIT 4

MATCH (catelyn:Character {name: "Catelyn"}),(drogo:Character{name:"Drogo"}) MATCH p=shortestPath((catelyn)-[INTERACTS\*]-(drogo)) RETURN p



MATCH (catelyn:Character {name: "Catelyn"}), (drogo:Character {name: "Drogo"}) MATCH p=allShortestPaths((catelyn)-[INTERACTS\*]-(drogo)) RETURN p



MATCH (c:Character) RETURN c.name AS character, size((c)-[:INTERACTS]-()) AS degree ORDER BY degree DESC

